

**IN THE CLAIMS:**

The text of all pending claims, (including withdrawn claims) is set forth below. Cancelled and not entered claims are indicated with claim number and status only. The claims as listed below show added text with underlining and deleted text with ~~striketrough~~. The status of each claim is indicated with one of (original), (currently amended), (cancelled), (withdrawn), (new), (previously presented), or (not entered).

Please CANCEL claims 2, 7 and 12 and AMEND claims 1, 3-6, 8-11 and 13-15 in accordance with the following:

1. (currently amended) A method for controlling a plurality of threads that perform parallel processing, the method comprising ~~the steps of:~~

monitoring a number of running threads performing parallel processing and a number of standby threads that are in a standby state; and ~~terminating~~

comparing the number of the standby threads in accordance with the a necessary  
number of the running threads at a predetermined time interval; and

terminating an amount of the standby threads exceeding the necessary number when the  
number of the standby threads is greater than the necessary number.

2. (cancelled)

3. (currently amended) The method according to claim[[ 2]]1,

wherein the ~~predetermined-necessary number refers to~~includes a maximum number of the running threads during a predetermined time period, and ~~the~~

wherein said comparing ~~step~~includes comparing the maximum number of the running threads and the number of the standby threads.

4. (currently amended) The method according to claim[[ 2]]1,

wherein the ~~predetermined-necessary number refers to~~includes an average number of the number of the running threads during a predetermined time period, and ~~the~~

wherein said comparing ~~step~~includes comparing the average number of the running threads and the number of the standby threads.

5. (currently amended) The method according to claim[[ 2]]1,  
wherein the ~~predetermined~~-necessary number ~~refers to~~includes a product obtained by multiplying the number of the running threads during a predetermined time period by a predetermined coefficient, and ~~the~~  
wherein said comparing step compares the product and the number of the standby threads.
6. (currently amended) A controller for controlling a plurality of threads that perform parallel processing, the controller comprising:  
a thread management table ~~for~~-storing thread information of the plurality of threads, wherein the thread information includes a number of running threads performing parallel processing and a number of standby threads that are in a standby state;  
a thread management circuit ~~for~~,requesting thread generation based on the number of the standby threads stored in the thread management table, ~~requesting thread generation and~~  
~~for~~-requesting a standby thread to run; and  
a ~~thread termination comparison~~ circuit ~~for terminating~~ comparing the number of the standby threads in accordance with a necessary number at a predetermined time interval; the number of the running threads and the number  
a termination circuit terminating an amount of the standby threads stored in the thread management table exceeding the necessary number when the number of the standby threads is greater than the necessary number.

7. (cancelled)

8. (currently amended) The controller according to claim[[ 7]]6, wherein the ~~predetermined~~-necessary number is a maximum value of the running threads during a predetermined time period.

9. (currently amended) The controller according to claim[[ 7]]6, wherein the ~~predetermined~~-necessary number is an average value of the running threads during a predetermined time period.

10. (currently amended) The controller according to claim[[ 7]] 6, wherein the ~~predetermined~~-necessary number is a product obtained by multiplying the number of the running threads during a predetermined time period by a predetermined coefficient.

11. (currently amended) A computer readable storage medium storing a program for controlling at least one processor to execute a plurality of threads that perform parallel processing, ~~wherein the program performs according to a method comprising the steps of:~~

monitoring a number of running threads performing parallel processing and a number of standby threads that are in a standby state; and ~~terminating~~

comparing the number of the standby threads in accordance with the a necessary number of the running threads at a predetermined time interval; and

terminating an amount of the standby threads exceeding the necessary number when the number of the standby threads is greater than the necessary number.

12. (cancelled)

13. (currently amended) The storage medium according to claim[[ 12]] 11, wherein the ~~predetermined~~-necessary number refers to a maximum number of the running threads during a predetermined time period, and ~~the~~

wherein said comparing ~~step~~ includes comparing the maximum number of the running threads and the number of the standby threads.

14. (currently amended) The storage medium according to claim[[ 12]] 11, wherein the ~~predetermined~~-necessary number refers to an average number of the number of the running threads during a predetermined time period, and ~~the~~

wherein said comparing ~~step~~ includes comparing the average number of the running threads and the number of the standby threads.

15. (currently amended) The storage medium according to claim[[ 12]] 11, wherein the ~~predetermined~~-necessary number ~~refers to~~ includes a product obtained by multiplying the number of the running threads during a predetermined time period by a predetermined coefficient, and ~~the~~

wherein said comparing ~~step~~ compares the product and the number of the standby threads.